AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method of preventing an attack on a network, wherein the attack comprises injecting a spurious transmission control protocol (TCP) segment into a TCP connection between a sender and a receiver, the method comprising the computer-implemented steps of:

receiving a TCP segment carrying a sequence value and an ACK value;

determining whether the ACK value is less than the difference of a next unacknowledged sequence value and a lesser of either (a) a total number of bytes sent in the TCP connection or (b) a maximum window size associated with the TCP connection; and

discarding the TCP segment when the ACK value is less than the difference of a next unacknowledged sequence value and the lesser of either (a) the total number of bytes sent in the TCP connection or (b) the maximum window size associated with the TCP connection.

2. (Original) A method as recited in Claim 1, wherein the steps are performed by an endpoint node acting as the receiver of data in the TCP connection.

3. (Original) A method as recited in Claim 1, wherein the steps are performed by a TCP application of an operating system of a network infrastructure element.

4. (Original) A method as recited in Claim 1, wherein the steps are performed by a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.

5. (Original) A method as recited in Claim 1, wherein the maximum window size comprises a maximum TCP sequence value window size that an endpoint node in the TCP connection can manage without regard to any change in current window size that either endpoint may establish during the TCP connection.

6. (Original) A method as recited in Claim 1,

wherein the determining step comprises determining whether the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window; and

wherein the discarding step comprises discarding the TCP segment when the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window.

7. (Currently amended) A method of preventing an attack on a network, wherein the attack comprises injecting a spurious transmission control protocol (TCP) segment into a TCP connection between a sender and a receiver the method comprising the computer-implemented steps of:

receiving a first TCP segment carrying a sequence value;

determining whether the sequence value is equal to a next expected sequence value;

when the sequence value is equal to a next expected sequence value, determining whether data carried in the first TCP segment overlaps data carried in one or more second TCP segments that were previously received in a re-assembly buffer; and

discarding the ~~one or more second~~ all TCP segments ~~from~~ that are in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

8. (Currently amended) A method as recited in Claim 7, ~~wherein the discarding step comprises discarding all TCP segments that are~~ further comprising storing the first TCP segment in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

9. (Original) A method as recited in Claim 7, wherein the data carried in the first TCP segment overlaps data carried in the one or more second TCP segments that were previously received in the re-assembly buffer when a first sum of a first sequence value and data length carried in the first TCP segment is less than a second sequence value carried in any of the second segments.

10. (Original) A method as recited in Claim 7, wherein the discarding step is performed when the first TCP segment completely overlaps any data segment previously received in the re-assembly buffer.

11. (Original) A method as recited in Claim 7, further comprising the step of sending an acknowledgment message that acknowledges data the sequence values of the first TCP segment.

12. (Original) A method as recited in Claim 7, wherein the steps are performed by an endpoint node acting as the receiver of data in the TCP connection.

13. (Original) A method as recited in Claim 7, wherein the steps are performed by a TCP application of an operating system of a network infrastructure element.

14. (Original) A method as recited in Claim 7, wherein the steps are performed by a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.

15.-17. (Canceled)

18. (New) A computer-readable tangible storage medium carrying one or more sequences of instructions for preventing an attack on a network, wherein the attack comprises sending a spurious transmission control protocol (TCP) segment with unwanted or spurious DATA, wherein the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform:

receiving a TCP segment carrying a sequence value and an ACK value;

determining whether the ACK value is less than the difference of a next unacknowledged sequence value and a lesser of either (a) a total number of bytes sent in the TCP connection or (b) a maximum window size associated with the TCP connection; and discarding the TCP segment when the ACK value is less than the difference of a next unacknowledged sequence value and the lesser of either (a) the total number of bytes sent in the TCP connection or (b) the maximum window size associated with the TCP connection.

19. (New) A computer-readable tangible storage medium carrying one or more sequences of instructions for preventing an attack on a network, wherein the attack comprises injecting a spurious transmission control protocol (TCP) segment into a TCP connection between a sender and a receiver, wherein the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform:

receiving a first TCP segment carrying a sequence value;

determining whether the sequence value is equal to a next expected sequence value;

when the sequence value is equal to a next expected sequence value, determining whether data carried in the first TCP segment overlaps data carried in one or more second TCP segments that were previously received in a re-assembly buffer; and

discarding the all TCP segments that are in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

20. (New) An apparatus for preventing an attack on a network, wherein the attack comprises sending a spurious transmission control protocol (TCP) segment with a spurious or unwanted DATA, comprising:

means for receiving a TCP segment carrying a sequence value and an ACK value;

means for determining whether the ACK value is less than the difference of a next unacknowledged sequence value and a lesser of either (a) a total number of bytes sent in the TCP connection or (b) a maximum window size associated with the TCP connection; and

means for discarding the TCP segment when the ACK value is less than the difference of a next unacknowledged sequence value and the lesser of either (a) the total number of bytes sent in the TCP connection or (b) the maximum window size associated with the TCP connection.

21. (New) An apparatus as recited in Claim 20, comprising an endpoint node acting as the receiver of data in the TCP connection.

22. (New) An apparatus as recited in Claim 20, wherein the means comprise a TCP application of an operating system of a network infrastructure element.

23. (New) An apparatus as recited in Claim 20, wherein the means comprise a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.

24. (New) An apparatus as recited in Claim 20, wherein the maximum window size comprises a maximum TCP sequence value window size that an endpoint node in the TCP connection can manage without regard to any change in current window size that either endpoint may establish during the TCP connection.

25. (New) An apparatus as recited in Claim 20,

wherein the determining means comprises means for determining whether the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window; and

wherein the discarding means comprises means for discarding the TCP segment when the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window.

26. (New) An apparatus for preventing an attack on a network, wherein the attack comprises sending a spurious transmission control protocol (TCP) segment with spurious or unwanted DATA, comprising:

a processor;

one or more stored sequences of instructions that are accessible to the processor and which, when executed by the processor, cause the processor to perform:

receiving a TCP segment carrying a sequence value and an ACK value;

determining whether the ACK value is less than the difference of a next unacknowledged sequence value and a lesser of either (a) a total number of bytes sent in the TCP connection or (b) a maximum window size associated with the TCP connection; and

discarding the TCP segment when the ACK value is less than the difference of a next unacknowledged sequence value and the lesser of either (a) the total number of bytes sent in the TCP connection or (b) the maximum window size associated with the TCP connection.

27. (New) An apparatus as recited in Claim 26, comprising an endpoint node acting as the receiver of data in the TCP connection.

28. (New) An apparatus as recited in Claim 26, wherein the steps are performed by a TCP application of an operating system of a network infrastructure element.

29. (New) An apparatus as recited in Claim 26, wherein the steps are performed by a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.

30. (New) An apparatus as recited in Claim 26, wherein the maximum window size comprises a maximum TCP sequence value window size that an endpoint node in the TCP connection can manage without regard to any change in current window size that either endpoint may establish during the TCP connection.

31. (New) An apparatus as recited in Claim 20,

wherein the determining step comprises determining whether the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window; and

wherein the discarding step comprises discarding the TCP segment when the ACK value is equal to an expected ACK value or a range of values less than an initial sequence value window..

32. (New) An apparatus for preventing an attack on a network, wherein the attack comprises injecting a spurious transmission control protocol (TCP) segment into a TCP connection between a sender and a receiver, the apparatus comprising:

means for receiving a first TCP segment carrying a sequence value;

means for determining whether the sequence value is equal to a next expected sequence value;

means for determining, when the sequence value is equal to a next expected sequence value, whether data carried in the first TCP segment overlaps data carried in one or more second TCP segments that were previously received in a re-assembly buffer; and

means for discarding the all TCP segments that are in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

33. (New) An apparatus as recited in Claim 32, further comprising means for storing the first TCP segment in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

34. (New) An apparatus as recited in Claim 32, wherein the data carried in the first TCP segment overlaps data carried in the one or more second TCP segments that were previously received in the re-assembly buffer when a first sum of a first sequence value and data length carried in the first TCP segment is less than a second sequence value carried in any of the second segments.

35. (New) An apparatus as recited in Claim 32, wherein the discarding means comprises means for discarding when the first TCP segment completely overlaps any data segment previously received in the re-assembly buffer.

36. (New) An apparatus as recited in Claim 32, further comprising means for sending an acknowledgment message that acknowledges data the sequence values of the first TCP segment.

37. (New) An apparatus as recited in Claim 32, comprising an endpoint node acting as the receiver of data in the TCP connection.

38. (New) An apparatus as recited in Claim 32, wherein the means comprise a TCP application of an operating system of a network infrastructure element.

39. (New) An apparatus as recited in Claim 32, wherein the means comprise a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.

40. (New) An apparatus for preventing an attack on a network, wherein the attack comprises sending a spurious transmission control protocol (TCP) segment with spurious or unwanted DATA, comprising:

a processor;

one or more stored sequences of instructions that are accessible to the processor and which, when executed by the processor, cause the processor to carry out the steps of:

receiving a first TCP segment carrying a sequence value;

determining whether the sequence value is equal to a next expected sequence value;

determining, when the sequence value is equal to a next expected sequence value, whether data carried in the first TCP segment overlaps data carried in one or more second TCP segments that were previously received in a re-assembly buffer; and

discarding the all TCP segments that are in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

41. (New) An apparatus as recited in Claim 40, further comprising instructions for storing the first TCP segment in the re-assembly buffer when the first TCP segment overlaps any data segment previously received in the re-assembly buffer.

42. (New) An apparatus as recited in Claim 40, wherein the data carried in the first TCP segment overlaps data carried in the one or more second TCP segments that were previously received in the re-assembly buffer when a first sum of a first sequence value and data length carried in the first TCP segment is less than a second sequence value carried in any of the second segments.

43. (New) An apparatus as recited in Claim 40, wherein the instructions for discarding comprise instructions for discarding when the first TCP segment completely overlaps any data segment previously received in the re-assembly buffer.

44. (New) An apparatus as recited in Claim 40, further comprising instructions for sending an acknowledgment message that acknowledges data the sequence values of the first TCP segment.

45. (New) An apparatus as recited in Claim 40, comprising an endpoint node acting as the receiver of data in the TCP connection.

46. (New) An apparatus as recited in Claim 40, wherein the instructions comprise a TCP application of an operating system of a network infrastructure element.

47. (New) An apparatus as recited in Claim 40, wherein the instructions comprise a TCP process, stack, adapter or agent hosted by or associated with an operating system of a personal computer, workstation or other network end station.